2011-02

A practical situation based agent architecture for social simulations

Alt, Jonathan K.

In M. K. M. Jakobson Gabriel; Endsley (Ed.), 2011 IEEE First International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)..

A Practical Situation Based Agent Architecture for Social Simulations

Jonathan K. Alt, Francisco Baez and Christian J. Darken

Abstract—**The concept of situation is central to the decision making processes of both human and software agents. The recognition of situation facilitates decision processes that ultimately result in action selection. Cognitive agent architectures that incorporate the concept of situation provide the opportunity for more sophisticated representations of human behavior and for more sophisticated decision support applications. This paper provides an overview of a general cognitive architecture for use in multi-agent simulation with the concept of situation central to the action selection and decision making process.**

Index Terms—**situation, learning, cognition**

I. Introduction

THE cognitive architecture proposed in this paper provides a minimalist approach for modeling human decision making based on the concept of situation. While multiple cognitive architectures exist in the literature, the framework proposed here seeks to incorporate the impact on relevant concepts from cognitive science, psychology, and social psychology in a relatively simple manner. The intent is to avoid a "kitchen" sink approach by identifying a framework to account for the influence of these notions using the smallest number of concepts and parameters possible. The prototype architecture provides a framework for experimentation with software agents for use in agent based social simulations with potential for the use of the architecture in conjunction with empirical data collection efforts. The need for an agent decision making architecture centered on the recognition of a given situation is highlighted by the literature on decision making and the need to reduce complex state spaces in agent environments [1], [2].

Agent architectures capable of recognizing relevant situations enable the use of algorithms such as reinforcement learning [3]. Reinforcement learning provides multiple techniques to enable software agents to select actions in given situations based on a reward policy specified by the modeler. The use of utility based rewards allows these policies to be tailored to the desired use case and role.

Manuscript received October 22, 2010.

Jonathan K. Alt is a student at the Naval Postgraduate School's Modeling, Virtual Environments and Simulation Institute (MOVES), Monterey, CA 93943 USA (831-656-7576; fax: 831-656-3084; e-mail: jkalt@nps.edu).

Christian J. Darken is an Associate Professor of Computer Science at the Naval Postgraduate School's MOVES Institute, Monterey, CA 93943 USA (e-mail: cjdarken@nps.edu).

Applications such as battlefield command and control systems and agent based social simulations require agents capable of allocating selective attention to relevant percepts in a given context. This combination of bottom up and top down processing in conjunction with the constraints of working memory facilitate the construction of a situation. The framework allows for the representation of human behavioral phenomena such as change blindness, where changes in a scene are not observed due to the effects of top down processing on selective attention. This framework also provides a mechanism for agents to participate in collective learning within a social network, to determine which agents to communicate with and to determine what messages should be attended to.

This paper reviews the concept of situation as defined by both the cognitive science and artificial intelligence literature. Next, the paper describes the proposed agent architecture and the utility-based action selection policy. Initial results of the prototype architecture in a benchmark environment are discussed. The paper ends with a discussion of planned future work.

II. Background

This section provides a brief overview of the concept of situation as used in this paper, cognitive architectures, cognitive social simulation, and reinforcement learning.

A. Situation

The term situation is typically used to refer to the current circumstances in which an entity finds itself. The situation can encompass the state of the external environment and the internal state of the entity itself. This term is often used in conjunction with a notion of awareness. Taken together, situation awareness is often defined as the "perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future [4]." This notion of situation is further decomposed into three levels: 1) perception, 2) comprehension, and 3) projection. Perception involves the detection of information from the environment. Comprehension involves the attribution of meaning from the perceived information relative to the individual's goals. Projection involves forecasting the impact of the individual's current comprehension of the perceived information on future states of the world relevant to their goals [5].

The term situation, as defined in relation to situation calculus, refers to an initial state, *s*, as well as the result of action, *a*, in state *s*, Result(*s*, *a*) [1]. A situation in this language is formed by the set of percepts that an individual senses from its environment, as constrained by the rules of the environment and the capability of its sensors. The agent's ability to use percepts to form and recognize the situation can be constrained by concepts such as working memory and selective attention [6]. A combination of bottom up processing, sensing of salient percepts from the environment, with top down processing of information, relevant to a given task or context, results in the formation and recognition of a situation [6]. Individuals or agents can then leverage the recognition of this situation to select appropriate actions given their goal set and prior knowledge. The impact of prior knowledge and experience on the ability to recognize situations relevant to the task at hand and quickly make decision has been examined extensively, forming the basis for recognition prime decision making [2].

B. *Cognitive Architectures*

Cognitive architectures provide a specification of the structural components of intelligent software agents of varying levels of sophistication, but all with a common goal of implementing a unified theory of cognition. The degree to which the architecture seeks to replicate what is known of human cognition varies by use case. A distinction is made in the literature between cognitive models, which are more narrowly scoped and seek to explain some phenomena, and cognitive architectures which seek to provide domain generic frameworks representative of the functional processes that occur within the human mind. Three broad categories of cognitive architectures have been identified by the National Research Council: symbolic, sub-symbolic (or connectionist), and hybrid [7]. Multiple cognitive architectures have been implemented and used in a variety of settings. Three of the more prominent cognitive architectures are ACT-R, SOAR, and CLARION [7].

ACT-R, whose development started in 1983, has traditionally focused on serving as a platform for research on cognition and representation of fundamental psychological processes [7]. ACT-R uses a combined form of symbolic and numerical representation with production firing based on log odds of success of a particular rule in a given situation [8]. ACT-R deals with the notion of working memory capacity through its use of declarative memory. Declarative memory decays as the size of the information pushed in increases, making it seem more closely tied to a capacity based representation of working [9]. ACT-R has been used in a number of applied settings to include the modeling of adversarial behavior [7].

The SOAR architecture has gone through eight major versions between 1982 and 2007, all maintaining a pure symbolic processing approach and using production rules for long-term knowledge representation [10]. The traditional SOAR, up through SOAR 8, consisted of symbolic long-term memory and symbolic short-term memory. The long-term memory represented knowledge as production rules. The short-term memory contained the agent's current assessment of the situation, based on perception and information from long-term memory. SOAR is characterized as taking a functional approach to the representation of working memory capacity. The general processing cycle in SOAR is to receive an input from perception causing changes in short-term memory, based on changes in short-term memory the goal of the agent is revisited, and operators are proposed and evaluated on their appropriateness to achieve the given goal based on the notion of a symbolic or numeric preference. Fixed decision procedures then select the appropriate operator, with mechanisms in place to accommodate conflicts should they arise. The actions associated with the chosen operator are executed by the rule based system with appropriate output passed to the environment [10]. Laird highlights extensions to the traditional SOAR in the latest version, SOAR 9, to provide capability for long-term memory representation, additional learning mechanism, and non-symbolic processing [10].

CLARION is a cognitive architecture consisting of four subsystems: the action-centered subsystem, the non-action centered subsystem, the motivational subsystem, and the meta-cognitive subsystem. Each subsystem provides two levels of knowledge representation, a top level for explicit knowledge representation and a bottom level for implicit knowledge representation. These bottom-up associations between action, state, and outcome inform action selection [9]. Interaction occurs between the two levels during action selection and learning. The action-centered subsystem controls all actions, external and internal to the agent. The non-action centered subsystem stores and maintains general knowledge. The motivational subsystem determines motivations for perception, action and cognition. The meta-cognitive subsystem controls the system of systems, providing central control [5]. The role of motivation and emotion in cognitive architectures will be discussed in greater detail below.

C. *Motivation and Emotion in Cognitive Architectures*

Sun points to evidence from social psychology supporting a dual view of human motivation, with implicit and explicit motivations playing a top down and bottom up role in the formation of intent [11]. The interplay between these explicit and implicit motivations as described by Sun allows for an implicit motivation, or need, to lead to a more explicitly stated motivational goal to satisfy the implicitly motivated need [12]. Sun provides a set of low (example: food, water, sleep etc) and high (example: social approval, social status, reciprocation etc) level primary drives as sources for implicit motivation [12]. The strength of these implicit motivations is determined by examining each in light of five considerations (proportional activation, opportunism, contiguity of action, interruption when necessary, and combination of preferences) with the result being the specification of an explicit motivational goal. This general idea is attractive, but a more compact means of addressing these fundamental motivations is employed here relying on Kenrick et al.'s renovation to the classic pyramid of needs [13].

The core of Maslow's theory of motivation rests on the notion that there are multiple independent fundamental motivational systems and that these motives are organized into a prioritized hierarchy [13]. The fundamental implicit motivations proposed by are from bottom to top: 1) immediate physiological needs, 2) self protection, 3) affiliation, 4) status/esteem, 5) mate acquisition, 6) mate retention, and 7) parenting. An important distinction to recognize is that the goals are overlapping, with lower level goals becoming activated once developed when appropriate situations arise, regardless of highest attained level of development. Kenrick states that a motivational system must contain "(a) a template for recognizing…environment threats or opportunities, (b) inner motivational/physiological states designed to mobilize relevant resources, (c) cognitive decision rules designed to analyze trade-offs inherent in various responses, and (d) a set of responses designed to respond to threats or opportunities represented by the environment inputs [13]." These functional requirements are contained in the meta-cognition module of the proposed architecture and reflect the central role of situation in cognition.

The role of emotion in the meta-cognitive processes is recognized as playing an important role in decision making and individual and social behavior [7]. At the most abstract level, emotions can be defined as mental states based on the individual or agent's current situation relative to its motivational goals and beliefs. Emotions express themselves in multiple means, but the primary focus of this research will be to represent their impact perception, cognition, and the appraisal process itself through a form of cognitive appraisal [7]. The notion is that the emotional state of the individual impacts its interpretation of the situation and subsequent decision making processes as well as its perception of subsequent information from the environment through the interaction of emotion with selective attention. Subsequent cognitive appraisals are also conducted from the view of the emotional state in which the agent resides when the appraisal begins. This aspect of affective processing remains to be fully explored and incorporated into cognitive architectures [7].

D. *Cognitive Social Simulations*

Cognitive architectures, described in some contexts as micro-level formal models, are simulation based models of human information processing often built to emphasize distinct aspects of cognition. Agent based models are tools for holistic analysis of systems, but require a reductionist approach in the development of micro-level behaviors for individual actors. Agents are intended to represent human behavior in the most simplified manner that is still useful [14]. Agent based social simulations represent human cognition at varying levels of sophistication [7], but typically adhere to the most rudimentary level of an agent as defined by Russell and Norvig. Summarized here, an agent senses information, or percepts from its environment, using sensors, updates its internal representation of the world, and selects actions based on this updated internal state [15]. Depending on the needed level of resolution the agent can represent either an individual or group of individuals. Sun points out that agent based social simulations and cognitive architectures have developed in relative isolation from each other, but that the use of appropriate cognitive architectures could benefit agent based social simulation by providing a realistic basis for the representation of individual agents [16]. While the potential for agent based social simulations and cognitive architectures to provide a multi-level examination of human behavior including the sociological and psychological perspectives respectively exists [8], the National Research Council is less clear on the use of cognitive architectures to represent group cognition [7]. Limited research has been conducted to develop cognitive architectures to address the representation of groups, this area has not been fully developed or applied broadly in social simulations [17].

E. *Reinforcement Learning*

Reinforcement learning provides a flexible tool to facilitate action selection in agent modeling across multiple domains. Utilizing these techniques, an agent can leverage the percepts from its environment received via organic sensors to select actions to execute in the environment via actuators [15]. Depending on the agent prototype chosen, the agent might make use of only the most recent percepts or an ordered sequence of percepts in order to assess the state of the environment, or situation. The agent uses these percepts to understand its current situation and identify the action choices relevant to the current situation. The basic elements of reinforcement learning (a policy, a reward function, a value function, and an optional model of the environment) allow the agent to identify how to map situation to actions [3]. The complexity of domains comprising the application areas for agents motivates the need for approaches to reinforcement learning that learn robust policies while efficiently utilizing computing resources.

Model free methods of reinforcement learning, such as Q-learning, provide general purpose methods of learning associations between rewards and actions [3]. Various methods exist to handle credit assignment and the problem of balancing exploration versus exploitation. The use of utility as a reward function provides the modeler with great flexibility in defining goals and objectives [15]. The concept of a state action pair is often used to describe the relationship between actions and the agent's current information regarding the state of the environment in which it is operating. The notion that equivalent states can be grouped into a set of situations has not been fully expanded in the literature on reinforcement learning. Reinforcement learning techniques have been successfully incorporated into existing cognitive architectures, but these architectures do not link the learning to the notion of a situation put forward in this research [16].

III. PROPOSED COGNITIVE ARCHITECTURE

This section provides a description of a practical cognitive architecture for use in social simulations in which the notion of a situation plays a central role.

A. Architectural Overview

A cognitive architecture provides structure to integrate cognitive models, models which attempt to account for functionality within the human brain, with a unified representation of cognition. The architecture presented here is based on the information processing model provided by Wickens et al. and is influenced by perceptual control theory [18]. The proposed cognitive architecture attempts to represent individual situation based cognition suitable for use in software agents intended to be integrated into social simulations. The architecture is presented in a general manner in this section.

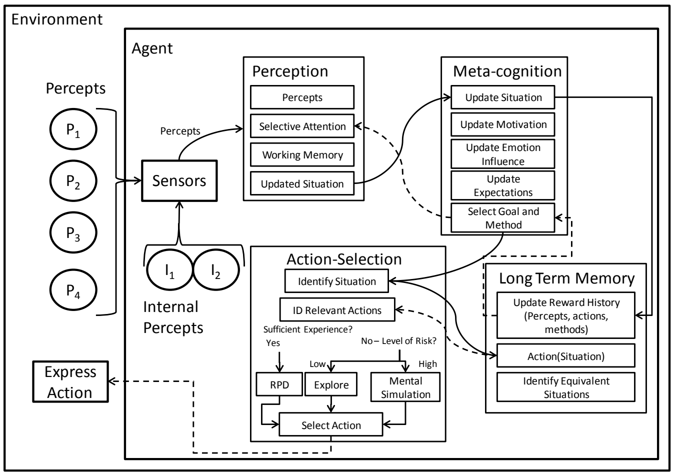

Figure 1. Situation based cognitive architecture.

Each of the functional components for the proposed architecture and their relationships to the whole will be briefly described prior to a more detailed description of each component in subsequent sections. The perception module, influenced by the goals of the agent from meta-cognition, receives and processes information from the environment in the form of percepts, subject to constraints on working memory and the allocation of selective attention influenced by elements of meta-cognition. The output of the perception module is a situation based on the most recent sensory information available. The meta-cognition module receives and processes the situation from the perception module, updating motivations, emotional state, expectations and ultimately outputting an updated goal state along with a top level method category to achieve that goal. The decision making module uses the updated situation, annotated with additional information from the meta-cognition module, to determine the method of action selection and ultimately the action to be expressed in the environment. The long-term memory module interacts with the other modules in the architecture providing a repository for long term goals, beliefs, values and interests, as well as reward histories, methods, and actions. This architecture distinguishes itself from previously proposed architectures for cognitive social simulation through the use of expectations in meta-cognition module, the incorporation of the notion of mental simulation and the central role of situation.

B. Perception: Working Memory, Selective Attention, and the Formation of Situation

The main function of the perception module is to form a situation constrained by the limits of working memory and informed by selective attention. Percepts arrive to the perception module via sensors that sense information from the environment and from the internal agent feedback mechanisms. Note that this architecture treats inter-agent communications through the receipt of information via percepts and the decision to communicate via action selection. Percepts are screened for relevance based on selective attention and if found relevant to the current situation are processed into working memory. Selective attention is driven by top down processing from the task and context [6]. Selective attention is influenced by the current motivations and emotions from the meta-cognition module. In communications selective attention is informed by information regarding the relationship with the other agent and notions such as trust. Working memory is limited to 7-10 percepts, the generally accepted limit [18]. The final set of percepts is considered a description of the current situation, considering both external perceptual information and information from the agent's internal state.

1. Percept received from sensors.
2. Percept is checked *for Relevancy(selective attention, percept)*; if percept is relevant it is passed to working memory.
3. If space is available in working memory percept is added; else oldest percept is dropped.
4. Current set of percepts is used to create a *Situation(percept1,..perceptN)* which is passed to the meta-cognition module for processing.

Figure 2. Steps in perceptual process.

C. Meta-cognition: Motivation, Emotion, and the Establishment of Goals and Rewards

The meta-cognitive module provides the agents top-down direction based on motivations and emotions elicited by a given situation input from the perception module. Meta-cognition is broadly defined as any cognitive process that monitors or controls other aspects of cognition or thinking about thinking [17]. Meta-cognition is described by Flavel as occurring in three phases: 1) meta-cognitive knowledge stores information regarding the environment, task, and known strategies; 2) meta-cognitive experience stores information describing previous means of achieving a given result; 3) meta-cognitive regulation describes the process of monitoring and controlling progress on cognitive tasks [19]. The meta-cognitive module also hosts the agent's information regarding the motivation of agent's behavior.

The input to the meta-cognition module is the most recent situation provided by the perception module. Using this updated situation the meta-cognition module conducts an update to determine which motivations are active and to assess the emotional impact of the new situation. The situation object in conjunction with information from long term memory is used to form expectations about likely future situations [20]. Goals and methods are selected using input on motivation, emotional state, and expectations in conjunction with long term memory. As a result of this step selective attention is updated and the updated situation and goal are passed to the action selection module.

1. Situation object is received from perception module.
2. *Motivation(situation)* and *Emotion(situation)* are determined.
3. A prediction of the next situation is accomplished based on *Expectation(situation, motivation, emotion)*.
4. The most urgent goal is chosen for action, *Goal(expectation, motivation, emotion)*.
5. Current situation and goal are passed to action selection module; update selective attention within perception module.

Figure 3. Steps in meta-cognitive process.

D. *Situation Based Action Selection: Learning, Recognition Prime Decision-making, and Mental Simulation*

Situation based action selection facilitates the reduction of the state space of the model through the notion of equivalent states being categorized as unique situations. For each unique situation there exists a set of relevant candidate actions. Actions have an associated activation level provided by a utility based reinforcement learning algorithm [3]. If the agent has enough experience, defined as a specified number of trials of each action, then the agent action selection is controlled by a softmax function, such as the Boltzman distribution, with a greedy setting, replicating recognition prime decision making [2]. If the agent has some level of experience in the situation then action selection can still be conducted using the softmax function, but with an exploratory setting. If the agent has no experience in the situation, then mental simulation is conducted, with the agent using available knowledge regarding the environment to project future states based on the actions currently available [2], [21]. An alternative to the case where sufficient experience is not present is to base the decision making mode on the risk level associated with the given situation. In this formulation, if the requisite experience to use recognition prime decision making is not present, when risk is low the agent simply uses the softmax function with an exploratory setting, while if risk is high the agent uses mental simulation.

1. Situation, method, and goal received from meta-cognition module.
2. RelevantActions(situation, method, goal) are identified.
3. DecisionMode(situation, actions, experience, risk) determines decision mode.
4. Action selected and expressed in the environment.

Figure 4. Steps in action selection process.

E. *Long-term Memory: Remembering the Situation*

Long-term memory stores information learned over time for future retrieval based on the situation. Reward histories from prior action selections as well as long term beliefs and issue stances are maintained in long term memory. Relevant actions for given situations as well as mappings of equivalent situations can be returned from long term memory based on need in a given situation.

IV. Prototype Implementation

This section describes the results of the prototype situation based cognitive architecture in a benchmark environment. The section provides an overview of the benchmark environment, a review of the functionality implemented in the current prototype, an overview of the experimental design, and analysis and results.

A. *Benchmark Environment*

The benchmark environment consists of a simple virtual environment with a text based interface modeled after the DikuMUD family of combat oriented MUD's. Player's of this type of game typically assume the role of a young adventurer with the goal of increasing the strength of an in game avatar [20]. This simple environment, implemented in Python, allows for the prototyping of the cognitive architecture described above.

B. *Current Architecture*

This description describes how the concepts described above are implemented in Python within the current architecture in the benchmark environment for use by the agent within the benchmark environment.

1) *Perception*

The perception module controls the agent's receipt of and use of information from the environment. Percepts are provided in the form of text based updates to state variables that describe the state of the environment from the agent's perspective [20]. Perception is constrained to those state variables that are collocated with the agent. Percepts describing the state of the agent are also provided. The environment provides four percept types: 'A' representing agent actions,' E' representing events, '+' representing the initialization of a time interval during which a variable was perceived, and ' –' representing the removal of a percept from the sensor range.

```
(20.0029997826 + location Red_Goblin79 The_Northern_Meadow4)
(23.7319998741 A w spock84)
(23.7319998741 E go spock84 west)
(23.7319998741 - place The_Northern_Meadow4)
```

Logical atoms are formed with percept type and the percept name, ignoring the time stamp and forming predicates with the result [20]. Working memory constrained by limiting the list of percepts to those that are no older than *t*. A example is shown below.

```
10 : ( percept goE spock84 south )
11 : ( percept sA spock84 )
12 : ( percept location- spock84 The_Northwestern_Meadow9 )
```

```
13 : ( percept place- The_Northwestern_Meadow9 )
14 : ( percept spock- spock84 )
15 : ( percept location+ spock84 The_Western_Meadow7 )
16 : ( percept Red_Goblin+ Red_Goblin79 )
```

A collection of percepts is used to identify the current situation through pattern matching and first order logic rules, with the situation then being declared as a current fact.

2) *Meta-cognition*

The meta-cognition module uses this current situation to determine the most relevant goal and method. This is again achieved through production system rules stored in long term memory that specify a goal for each situation and some number of methods to achieve that goal. In the case where multiple methods can be used to achieve the goal, reinforcement learning is used to select the method for execution. The particular method of reinforcement learning used here is described in the subsequent discussion of action selection. A sample rule is shown below.

```
(defrule FindObject
   (CurrentGoal ?g)
   (goalProperty ?g name FindObject)
   (percept spock+ ?userID)
   (percept location+ ?userID The_Southern_Meadow5)
=>
   (assert(action e))
```

3) *Long-term memory*

Long term memory contains the reward histories supporting the reinforcement learning algorithms, situation to action relationships, and top level rules. Meta-cognitive rules, situation to action mappings, and memory decay parameters are provided to the agent prior to run time. The firing time of each rule is recorded along with the time at which utility was received by the agent as shown below.

```
firing times [5.25899982452, 9.6609997749299996,
19.042999982800001]
utilities [(1.0, 14.263999939)]
```

4) *Action selection*

Given the goal and method based on the current situation from the perception module, the action selection module implements a form of recognition prime decision making. For the given situation, as represented by the goal and method, the agent selects from a subset of possible actions, constrained by the situation. The agent selects an action for expression in the environment using utility based reinforcement learning. Point utility is a real number (approximated in implementations by a floating-point number) that represents the agent's degree of contentment with the conditions in its environment at a specific point in time. Since the agent's knowledge of the current status of the environment comes only via percepts, and so can change only when a percept is received, we define a point utility value for each percept received. This value depends only upon that percept and the state at that time, as the state summarizes all past percepts. The point utility function is thus defined as $u: S, P \rightarrow \square$.

A single action will generally affect more than one point utility value. Therefore, it is important to aggregate utility in order to capture the effects of an action on point utility values received over time. The traditional aggregation method is to form the exponential moving average of the point utility values. Let p_i be the percept sequence, and t_i the sequence of times at which the percepts arrive. Let s_i be the corresponding sequence of states. Then the corresponding sequence of point utility values is $u_i=u(s_i,p_i)$. Given the choice of exponential base $0<\lambda<1$, the exponential moving average of the sequence starting at time t is,

$$\bar{q}(t) = \sum_i \lambda^{t_i - t} u_i \Theta(t_i - t) \tag{1.1}$$

,where Θ is the unit step function, which is zero when its argument is negative and one otherwise.

Clearly the expected future aggregate utility of an action in a particular situation must be an important factor in any decision to select it. The obvious estimator of the expected future aggregate utility of an action is the average of the aggregate utility received when the action was taken in the past. Let a_k be the action selected in situation σ_k at time t_k. Then the aggregate utility actually received after this action is given by $\bar{q}(t_k)$. Let define $t(\sigma, a)$ to be the set of all times at which action a was taken in situation σ. Then we will take the estimator of the expected future aggregate utility of action a in situation σ to be

$$\hat{Q}(\sigma, a) = \sum_{t \in t(a)} \frac{q(t)}{|t(\sigma, a)|} \tag{1.2}$$

, where $|t(\sigma,a)|$ $|t(\sigma,a)|$is the number of elements in the set $t(\sigma,a)$ $t(\sigma,a)$. $\hat{Q}$ is an estimator of the Q function typically defined in reinforcement learning *in the special case that the set of situations is identical to the set of individual states.*

Action selection is tricky for learning agents. To have the best possible chance at finding an optimal action selection policy, they must walk a line between adequate exploration of underused actions and exploitation of actions that have produced good results in the past. The most common approaches used in reinforcement learning, namely the ε-greedy method or the softmax (Boltzmann distribution) method are applicable here.

$$P_i = \frac{e^{\hat{Q}(\sigma,a)_i/\tau}}{\sum_j e^{\hat{Q}(\sigma,a)_j/\tau}} \tag{1.3}$$

The action i with the greatest expected utility, P, is selected. Note that in this case the temperature, τ, is used to control the level of exploration and exploitation.

C. *Experiment one*

The experiment will examine a simple agent seeking to find a randomly generated moving object within a 3 x 3 domain hosted on a MUD type server described above as an example of recognition primed decision making. Once the object is identified the agent remains in place until the object has moved, then continues to pursue the object as it continues its random movement. In this simple case the agent is asked to learn which of the four policies maximize its overall utility over the course of a set period. Each course of action (COA) is distinct. In COA 1, the agent patrols the perimeter of the grid-world and receives a reward for the detection of a single object. COA 2 differs from COA 1 only in the timing of the reward, which occurs following the sequential detection of two objects.

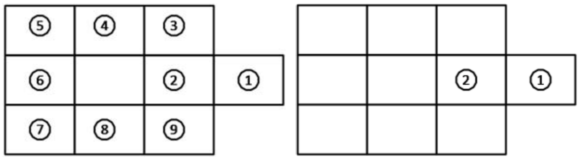

Figure 5. Search patterns COA1&2 left, COA3&4 right.

In COA 3, the agent remains in its current location until the domain generates a co-located object, at which time it receives a reward. COA 4 is identical to COA 3 with the exception of the requirement to identify 2 objects, as in COA 2. The reward for a single object, COA 1 and COA 3, is 1.0 and the reward for the sequential detection of two objects, COA 2 and COA 4, is 2.0. The agent has the opportunity to select one of the four possible COA's each time it finds itself in the situation of having just obtained a goal.

The agent in this case must learn which of the four COA's available should be chosen in the given situation. In this simple case the focus is on the recognition prime decision making component of the cognitive architecture. Two settings of the temperature parameter, 0.333 and 0.667, that controls the level of exploration and exploitation of the learning agent through the Boltzmann distribution were explored. Each case was allowed to run for 10,800 seconds resulting in 113 decision points for the medium temperature case and 127 decision points for the low temperature case. The low temperature case is equivalent to the notion of recognition prime decision making as a form of greedy reinforcement learning executed when the agent has sufficient experience in the environment. The medium temperature case might be used when the agent has some level of knowledge regarding the situation and the risk is perceived as relatively low.

Even with this midrange exploration setting, however, the expected utility of each COA begins to separate following approximately 80 decision points, and the rank order of the COA's does not change following the 69th decision point. Note that by the end of the run the activation levels for COA 2, actively searching for two objects, and COA 3, waiting for a single object, have separated themselves from COA 1 and COA 4 by over 0.1. The average activation level mirrors the rank order of the figure 6 (COA2 = 0.33, COA3=0.24, COA4= 0.19, COA1=0.14). The agent's performance in the environment as measured by time to detect an object did not improve with each successive attempt and in fact varied greatly over the latter portion of the run.

In the low temperature case, the expected utility of the COAs shows greater variability in the first 50 decisions than in the later portions of the run. Interestingly, COA 4 in this case does not show any change in expected utility until approximately decision point 50 as well. The ranking of the courses of action by expected utility becomes stable earlier in this case than the medium temperature case (~DP 50 as opposed to DP70), but late in the run COA 4 crosses over with COA 1 in ranking. The end of run ranking is the same as in the previous case. The average expected utility corresponds to the ranking seen above with the exception of COA1 and COA 4 (COA2 = 0.343, COA3=0.227, COA4= 0.093, COA1=0.193).

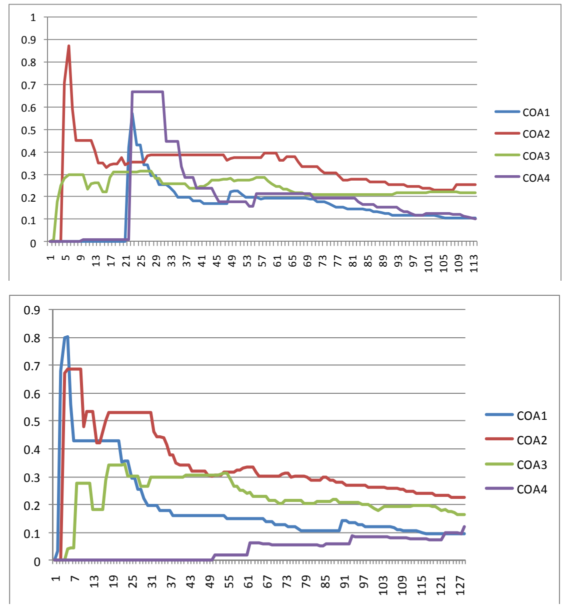

Figure 6. Expected utility for each course of action at sequential decision points for temp=0.667 (top) and temp=0.333 (bottom).

From the perspective of representing recognition prime decision making with reinforcement learning, the results of the bottom chart illustrate how, with some minimal level of experience in the situation and environment, in this case approximately 40 encounters with the situation, the agent is able to correctly choose the course of action with the greatest expected utility. In the case where the agent does not have sufficient experience and risk is perceived as low, a medium to high temperature setting allows the agent to explore for a greater period of time, approximately 40 exposures to the situation prior to the dominant COAs consistent selection. The next section will describe the use a dynamic temperature schedule in the same task.

D. *Experiment two*

The second experiment explores the use of a dynamic temperature during runtime, controlling the balance of exploration and exploitation based on the updated situation as described by percepts from the environment. The notion of controlling the temperature dynamically can facilitate a more robust representation of recognition prime decision making. Initially a novice agent will need to explore the environment to gain some level of experience, but once a sufficient level of experience is obtained the agent will need to exploit the knowledge it has learned about the environment. This section explores the use of time to control the temperature.

The simplest possible method of illustrating the concept of a dynamic temperature is through the use of a decay function driven by the current simulation time. The general form of this simple approach is shown below.

$$\tau_{new} = \frac{\tau_{initial}}{1 + t/t_e} \quad (1.4)$$

The new temperature, τ_{new}, is driven by a modeler driven half life specified by t_e with t being the current simulation time. In this case the simulation was run for 5000 seconds, with a half life specified at 2500 seconds and an initial temperature of 1.0. The resulting expected utilities of each course of action as well as the temperature over time are shown below.

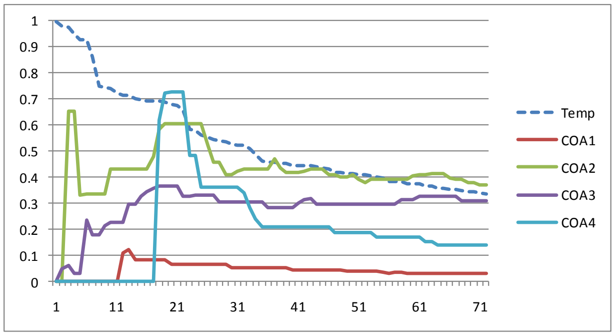

Figure 7. Expected utility of each course of action with dynamic temperature.

In this case, the agent converges to the preferred course of action by approximately the 20th decision point. This illustrates that even with a simple time based approach for dynamically adjusting the temperature, learning can be facilitated with fewer encounters with the decision point than with the fixed cases above.

V. Conclusions and Future Work

This paper introduced a situation based cognitive architecture for use in social simulation that builds on the concepts of perception, meta-cognition, action-selection, and long term memory. The proposed architecture imposes constraints on information processing representative of those that exist in humans. The perception module incorporates notions of working memory and selective attention. Meta-cognition is influenced by motivation and emotion. The action selection module incorporates concepts inspired by recognition prime decision making and the consideration of risk. Central to each process is the notion of situational relevancy.

Experimental results from the current prototype cognitive architecture were presented for a simple text based environment as a proof of principle. A utility based reinforcement learning method that places the notion of situation in a central role was introduced and results examined as a means of representing recognition prime decision making. A simple means of dynamically adjusting the balance between exploration and exploitation as a function of time was also introduced.

Future work will seek to implement the impact of emotion, motivation, and expectation on the selection of methods and goals in the meta-cognition module. The action selection module will be expanded to include selection of decision mode as during runtime allowing the agent to dynamically shift from recognition prime decision making to a more exploratory mode or to mental simulation based on the situation. The use of other elements of the situation to control the level of exploration and exploitation will be further explored. The incorporation of the architecture into a target social simulation, the Cultural Geography model, is ongoing [22]. The uses of cognitive architectures that leverage a robust notion of situation within a social setting have yet to be fully explored [16]. Ongoing efforts will explore the concept of trust formation and communications through the use of these technologies.

References

[1] S. Russel and P. Norvig, *Artificial intelligence: A Modern Approach*, Second. Prentice Hall, 2003.

[2] G. Klein, *Sources of Power: How People Make Decisions*. Massachusetts Institute of Technology, 1998.

[3] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998.

[4] M. Endlsey, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors*, vol. 37, no. 1, pp. 32-64, 1995.

[5] M. R. Endsley and E. S. Connors, "Situation awareness: State of the art," in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–4, 2008.

[6] J. Anderson, *Cognitive Psychology and Its Implications*, 6th ed. New York: Worth Publishers, 2005.

[7] National Research Council, *Behavioral Modeling and Simulation: From Individuals to Societies*. National Academies Press, 2008.

[8] R. Sun, *The Importance of Cognitive Architectures: An Analysis Based on CLARION*. Rensselaer Polytechnic Institute, 2006.

[9] N. A. Taatgen and J. R. Anderson, "Constraints in cognitive architectures," *Cambridge handbook of computational psychology*, pp. 170–185, 2008.

[10] J. E. Laird, "Extending the Soar cognitive architecture," in *Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, 2008.

[11] R. Sun, "The motivational and metacognitive control in CLARION," *Integrated models of cognitive systems. New York, NY: Oxford UP*, 2007.

[12] R. Sun, "Motivational representations within a computational cognitive architecture," *Cognitive Computation*, vol. 1, no. 1, pp. 91-103, 2009.

[13] D. T. Kenrick, V. Griskevicius, S. L. Neuberg, and M. Schaller, "Renovating the Pyramid of Needs: Contemporary Extensions Built Upon Ancient Foundations," *Perspectives on Psychological Science*, vol. 5, no. 3, pp. 292-314, 2010.

[14] N. Gilbert and P. Terna, "How to build and use agent-based models in social science," *Mind & Society*, vol. 1, no. 1, pp. 57–72, 2000.

[15] S. Russel and P. Norvig, *Artificial intelligence: A Modern Approach*, Second. Prentice Hall, 2003.

[16] R. Sun, "Cognitive social simulation incorporating cognitive architectures," *IEEE Intelligent Systems*, pp. 33-39, 2007.

[17] D. N. Davis and V. M. Venkatamuni, "A "Society of Mind" Cognitive Architecture Based on the Principles of Artificial Economics," *in International Journal of Artficial Life Research*, 2009.

[18] C. Wickens and J. Hollands, *Engineering Psychology and Human Performance*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2000.

[19] J. Flavell, "Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry," *American Psychologist*, vol. 34, pp. 906-911, 1979.

[20] C. Darken, "Towards learned anticipation in complex stochastic environments," *Proc. Artificial Intelligence for Interactive Digital Entertainment 2005*.

[21] D. Kunde and N. P. S. (US), *Event prediction for modeling mental simulation in naturalistic decision making*. Citeseer, 2005.

[22] J. Alt, L. Jackson, D. Hudak, and S. Lieberman, "The Cultural Geography Model: Evaluating the Impact of Tactical Operational Outcomes on a Civilian Population in an Irregular Warfare Environment," *Journal of Defense Modeling and Simulation*, In press. 2010.